FORMAS LIBRES (I)

CURVAS NURBS

por

SONIA LUISA RUEDA PÉREZ



CUADERNOS
DEL INSTITUTO
JUAN DE HERRERA
DE LA *ESCUELA DE*
ARQUITECTURA
DE MADRID

3-78-03

FORMAS LIBRES (I)

CURVAS NURBS

por

SONIA LUISA RUEDA PÉREZ

CUADERNOS
DEL INSTITUTO
JUAN DE HERRERA
DE LA *ESCUELA DE*
ARQUITECTURA
DE MADRID

3-78-03

C U A D E R N O S

D E L   I N S T I T U T O

J U A N   D E   H E R R E R A

NUMERACIÓN

3 Área

78 Autor

03 Ordinal de cuaderno (del autor)

TEMAS

1 ESTRUCTURAS

2 CONSTRUCCIÓN

3 FÍSICA Y MATEMÁTICAS

4 TEORÍA

5 GEOMETRÍA Y DIBUJO

6 PROYECTOS

7 URBANISMO

8 RESTAURACIÓN

0 VARIOS

Formas libres (I)
Curvas NURBS

Instituto Juan de Herrera
Escuela Técnica Superior de Arquitectura de Madrid
Gestión y portada: Alba Peña Fernández
CUADERNO 456.01 / 3-78-03
ISBN-13 (obra completa): 978-84-9728-538-4
ISBN-13: 978-84-9728-539-1
Depósito Legal: M-19075-2015

Índice general

Índice de figuras



Figura 1: Parque del Milenio, Chicago. F.O. Gehry, 2004

Las curvas del diseño asistido por ordenador (CAD, *Computer Aided Design*), son las curvas NURBS (de *Non uniform rational B-spline*). Son las curvas que se utilizan para crear formas libres y evolucionaron a partir de las curvas de Bézier (De Casteljau 1959, Bézier 1962) y de las curvas B-spline (*Basis Spline*).

Las curvas NURBS están definidas a partir de un conjunto de puntos de control. Hay dos técnicas diferentes para crear curvas a partir de puntos de control.

- Interpolación: Dado un conjunto de puntos ordenado, se genera una curva suave que pasa por todos esos puntos.

- Aproximación: Se traza un polígono de control que une los puntos ordenados y se busca una curva que aproxime la forma indicada por el polígono de control.

Las curvas NURBS se obtienen mediante un proceso de aproximación.

Estos apuntes están especialmente dirigidos a alumnos del Taller Experimental II: Proyectos de Arquitectura Paramétrica, y alumnos del Grado en Fundamentos de la Arquitectura, interesados en profundizar en las herramientas matemáticas utilizadas para crear curvas de forma libre (en Grasshopper y Rhinosceros). En la medida de lo posible, se ha intentado suavizar el rigor matemático, utilizando sólo el nivel necesario para trasmitir las posibilidades que ofrecen las curvas NURBS al diseñador. Se ha puesto énfasis en conectar el uso de Grasshopper con las matemáticas que soportan este programa de ordenador.

`Grasshopper` es un componente de Rhinosceros 3D que se ha popularizado por su capacidad para generar diseño paramétrico. Es un editor gráfico de algoritmos que no requiere conocimientos de programación, pero permite al diseñador la construcción de generadores de formas tanto simples como asombrosas.

Este es el primer volumen de la serie:

- FORMAS LIBRES I: Curvas NURBS.
- FORMAS LIBRES II: Superficies NURBS.

1 Curvas de Bézier

Para definir una curva de Bézier partimos de $n+1$ puntos $B_0, \dots B_n$ planos (en $\mathbb{R}^2$) o espaciales (en $\mathbb{R}^3$) que llamaremos vértices o puntos de control. El polígono de control es el polígono que obtiene tras unir el punto B_i con el B_{i+1}, $i = 0, \dots, n-1$ mediante un segmento, que podemos parametrizar por

$$B_i^1(t) = (1-t)B_{i+1} + tB_i, \;\; t \in [0,1].$$

Ejemplos 1.0.1. 1. *El polígono de control determinado por los puntos* $B_0 = (1,1)$, $B_1 = (3,5)$, $B_2 = (5,3)$, $B_3 = (7,4)$ *es el de la figura 1.1. Las parametrizaciones de los segmentos son*

$$B_0^1(t) = (1{+}2t, 1{+}4t), \;\; B_1^1(t) = (3{+}2t, 5{-}2t), \;\; B_2^1(t) = (5{+}2t, 3{+}u), \;\; t \in [0,1].$$

2. *Para puntos* $B_0 = (1,1,0)$, $B_1 = (1,3,5)$, $B_2 = (5,-1,3)$, $B_3 = (3,4,1)$ *en* $\mathbb{R}^3$ *las parametrizaciones de los segmentos del polígono de control (en la figura 1.1) son:*

$$B_0^1(t) = (1, -2t+3, 5-5t), \;\; B_1^1(t) = (-4t+5, 4t-1, 2t+3),$$
$$B_2^1(t) = (2t+3, -5t+4, 2t+1), \;\; t \in [0,1].$$

1.1. Algoritmo de De Casteljau

Una forma sencilla de comprender cómo se obtiene la parametrización de una curva de Bézier, a partir de los vértices del polígono de control, es el algoritmo de De Casteljau, para 4 vértices o puntos de control B_0, B_1, B_2, B_3. Sea t un parámetro en el intervalo $[0,1]$. Los pasos del algoritmo son los siguientes:

1. Interpolar linealmente pares de puntos consecutivos

$$B_0^1(t) = (1-t)B_0 + tB_1,$$
$$B_1^1(t) = (1-t)B_1 + tB_2,$$
$$B_2^1(t) = (1-t)B_2 + tB_3,$$

obteniendo la parametrización de las rectas del polígono de control.

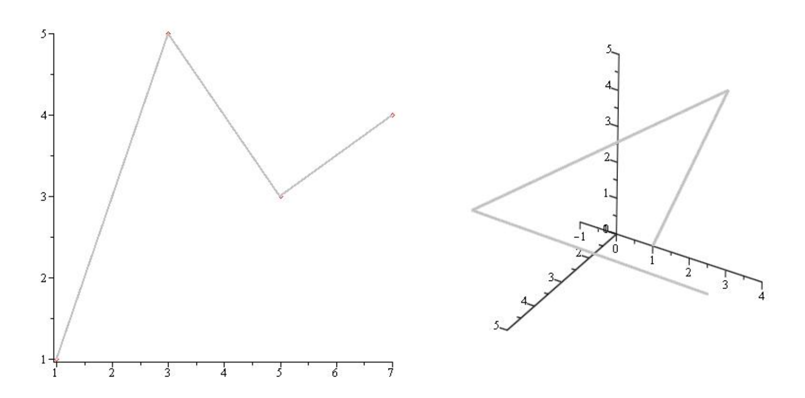


Figura 1.1: Polígono de control determinado por 4 puntos en $\mathbb{R}^2$ (izq) y $\mathbb{R}^3$ (dcha)

2. Interpolar linealmente pares de puntos genéricos consecutivos $B_0^1(t)$, $B_1^1(t)$, $B_2^1(t)$, para un mismo valor del parámetro t

$$\begin{aligned} B_0^2(t) &= (1-t)B_0^1(t) + tB_1^1(t), \\ B_1^2(t) &= (1-t)B_1^1(t) + tB_2^1(t), \end{aligned}$$

   obteniendo parametrizaciones de curvas de grado dos.

3. Interpolar linealmente los puntos genéricos de las curvas de grado 2, para un mismo valor del parámetro t

$$\begin{aligned} \mathcal{P}(t) &= (1-t)B_0^2(t) + tB_1^2(t), \\ &= (1-t)^3 B_0 + 3(1-t)^2 t B_1 + 3(1-t)t^2 B_2 + t^3 B_3. \end{aligned}$$

   obteniendo la parametrización $\mathcal{P}(t)$ de la curva de Bézier, cuyas componentes son polinomios de grado 3.

Ejemplo 1.1.1. *Calculamos con Maple, siguiendo este algoritmo, la parametrización de una curva de Bezier de grado* 3 *a partir de cuatro puntos* $B_0 = (0,0)$, $B_1 = (1,3)$, $B_2 = (5,4)$, $B_3 = (7,-1)$. *Calculamos la parametrización de las rectas del polígono de control,* $B_0^1(t)$, $B_1^1(t)$, $B_2^1(t)$ *que podemos ver como tres puntos genéricos (parametrizaciones de curvas de grado* 1*). Unimos estos tres puntos mediantes rectas* $B_0^2(t)$, $B_1^2(t)$ *obteniendo dos nuevos puntos genéricos (parametrizaciones de curvas de grado* 2*) y finalmente, la curva de Bézier viene dada por la recta que une dichos puntos, una parametrización dada por polinomios de grado* 3

$$\mathcal{P}(t) = \left(3t + 9t^2 - 5t^3, 9t - 6t^2 - 4t^3\right).$$

Representamos la curva junto a su polígono de control, en la figura 1.2.

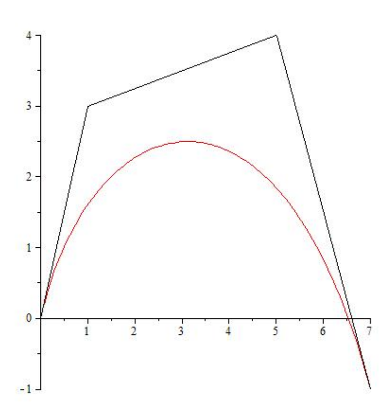


Figura 1.2: Polígono de control de una curva de Bézier de grado 3

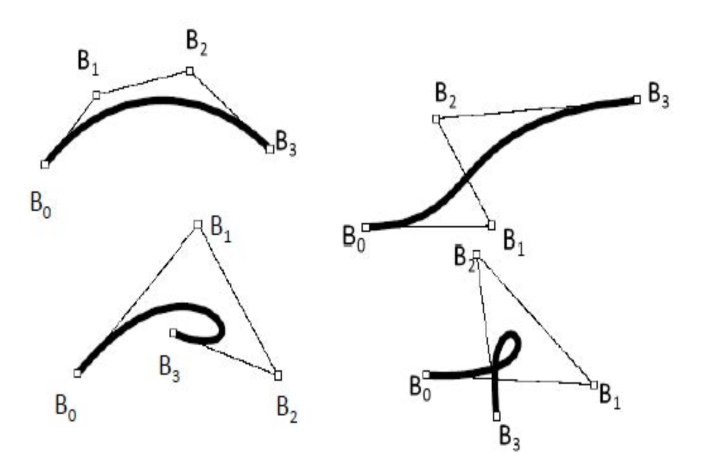


Figura 1.3: Curvas de Bézier de grado 3

Conocidas las propiedades básicas de las curvas de Bézier un usuario puede predecir (con un poco de práctica) la forma de una curva generada por un polígono de control dado (véase la figura 1.3).

1.2. Definición y propiedades

La parametrización que proporciona el algoritmo de De Casteljau se puede expresar mediante una fórmula en la base de los polinomios de Bernstein. Dados $n+1$ puntos de control $B_0, B_1, \ldots, B_n$, definimos una curva de Bézier mediante una para-

metrización, dada por polinomios de grado n

$$\mathcal{P}(t) = \sum_{i=0}^{n} B_i J_{n,i}(t), \;\; 0 \leq t \leq 1$$

$$\text{con polinomio de Bernstein } J_{n,i}(t) = \begin{pmatrix} n \\ i \end{pmatrix} t^i(1-t)^{n-i}.$$

La primera propiedad, fácilmente deducible de la parametrización, es que la curva de Bézier contiene a los puntos inicial y final del polígono de control

$$B_0 = \mathcal{P}(0) \text{ y } B_n = \mathcal{P}(1).$$

1.2.1. Vectores tangentes y derivadas

El vector tangente a la curva de Bézier en el punto $\mathcal{P}(t)$ viene dado por su primera derivada $\mathcal{P}'(t)$ [GAS]

$$\mathcal{P}'(t) = \sum_{i=0}^{n} B_i J'_{n,i}(t).$$

Comprobamos fácilmente que los vectores tangentes en los puntos inicial y final de la curva de Bézier tienen la misma dirección que el primer y último segmento del polígono de control, respectivamente (véase la figura 1.3).

$$\mathcal{P}'(0) = n(B_1 - B_0),$$
$$\mathcal{P}'(1) = n(B_n - B_{n-1}).$$

Por otra parte, la segunda deriva $\mathcal{P}''(t)$ está relacionada con la curvatura [GAS]. En los puntos inicial y final de la curva, la curvatura depende de los tres puntos de control más cercanos (de las dos aristas más cercanas del polígono de control).

$$\mathcal{P}''(0) = n(n-1)(B_0 - 2B_1 + B_2),$$
$$\mathcal{P}''(1) = n(n-1)(B_n - 2B_{n-1} + B_{n-2}).$$

Ejemplo 1.2.1. *Volvamos al caso de un curva de Bézier con 4 puntos de control. Derivando la parametrización obtenida en la Sección 1.1 obtenemos una interesante relación*

$$\begin{aligned} \mathcal{P}'(t) =& -3(1-t)^2 B_0 + 3(-2t(1-t) + (1-t)^2)B_1 + 3(-t^2 + 2t(1-t))B_2 + 3t^2 B_3 \\ =& \cdots = 3(B_1^2(t) - B_0^2(t)). \end{aligned}$$

Es decir, el vector tangente a la curva de Bézier en el punto $\mathcal{P}(t)$ es proporcional al segmento $B_1^2(t) - B_0^2(t)$

Si la curva de Bézier de grado 3 es espacial, los tres primeros puntos de control B_0, B_1, B_2 determinan el plano osculador en B_0 y los tres últimos B_1, B_2, B_3 determinan el plano osculador en B_3.

1.2.2. Envolvente convexa

Intuitivamente hablando la envolvente convexa de un conjunto de puntos en el plano se obtiene como sigue. Colocamos alfileres sobre dichos puntos y soltamos una goma a su alrededor, el polígono que forma la goma es la envolvente convexa. Con puntos espaciales podemos pensar en envolverlos con una membrana elástica. En el caso plano obtenemos un polígono convexo y en el espacial un poliedro convexo. En ambos casos, ser convexo significa que el segmento que une dos puntos cualesquiera está contenido en la envolvente.

Una curva de Bézier está contenida en la envolvente convexa de sus puntos de control. Esto se deduce de su construcción mediante el Algoritmo de De Casteljau, que realiza iteradamente interpolación lineal. Este hecho aporta control sobre el diseño. Observamos además, que las curvas de Bézier tienen precisión lineal, en el sentido de que si los puntos de control están alineados, la curva de Bézier está contenida en la recta que determinan. Véase el ejemplo de la figura 1.5.

1.2.3. Otras propiedades y limitaciones

Las curvas de Bézier son invariantes por transformaciones afines. Esto significa que la imagen mediante una transformación afín de una curva de Bézier coincide con la curva de Bézier de los transformados de los puntos de control. Ejemplos de transformaciones afines son las isometrías (traslaciones, rotaciones, simetrías), semejanzas (composición de isometrías con homotecias) y proyecciones (véanse [CLl] y [PAHK]).

Aunque sirven para modelar un amplio conjunto de curvas, las curvas de Bézier presentan dos limitaciones importantes:

1. Un número de puntos de control elevado produce curvas que se asemejan menos al polígono de control, el diseñador pierde la intuición de la curva que va a producir.

2. Cada punto de control tiene un efecto global sobre la curva de Bézier obtenida, ya que en la definición está multiplicado por un polinomio de Bernstein, con dominio en todo el intervalo $[0, 1]$. Por lo tanto, si desplazamos un punto o incluimos un nuevo punto, cambia la forma de la curva entera.

1.2.4. Componentes Grasshopper

Esta primera aproximación a las curvas NURBS permite al alumno empezar a comprender el funcionamiento real de algunos de los componentes más importantes de Grasshopper. Dada una lista P de puntos de control, el componente `PLine` dibuja el polígono de control que determinan. Para dibujar la curva de Bézier utilizamos

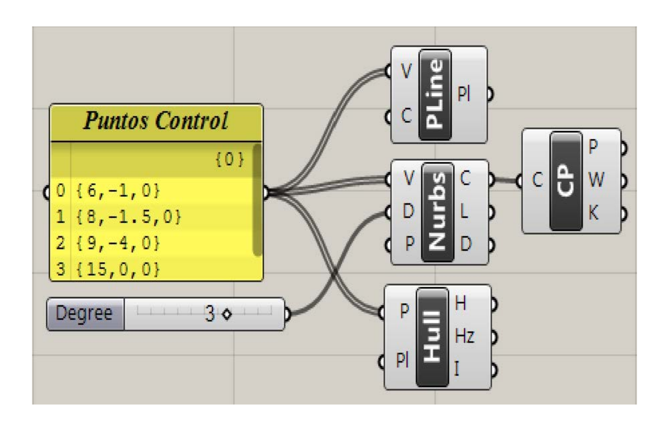


Figura 1.4: Componentes `PLine`, `Nurbs`, `Hull` y `CP` de Grasshopper

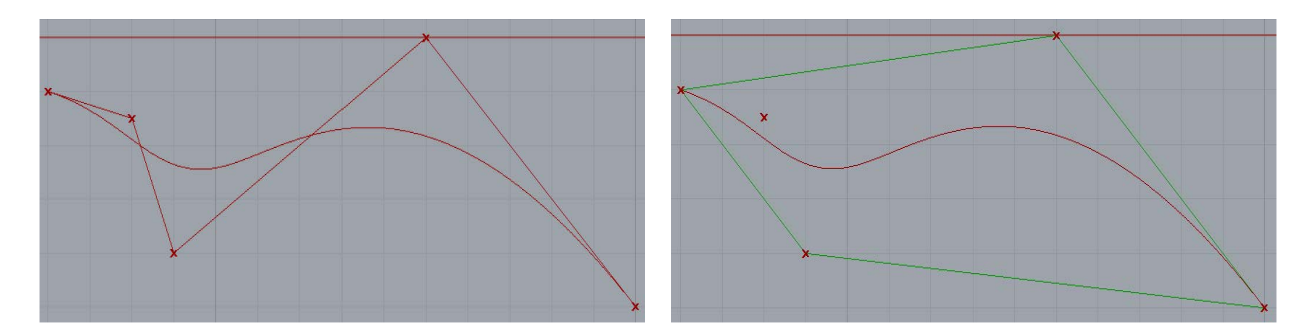


Figura 1.5: (Dcha) Polígono de control de una curva de Bézier. (Izq) Envolvente convexa de los puntos de control de una curva de Bézier plana

el componente `Nurbs` con grado D igual al número de puntos de control menos uno (véase la figura 1.4). En general `Nurbs` dibuja curvas B-spline como veremos más adelante.

El componente `Hull` de Grasshopper devuelve la envolvente convexa de un conjunto de puntos del plano. En la figura 1.5 se puede ver una curva de Bézier de grado 4, creada con `Nurbs`.

Dada una curva cualquiera definida en Grasshopper, podemos visualizar sus puntos de control utilizando `CP`. Este componente devuelve además la lista de pesos W y el vector de nudos K, como veremos posteriormente.

2 *Curvas B-spline*

Una curva definida a partir de un polígono de control depende de la estrategia de interpolación o aproximación que se utilice para construir la curva, depende de la base de funciones que se utilice para este objetivo. Utilizando los polinomios de Bernstein aparecen dos limitaciones importantes, el grado de la curva obtenida es siempre el número de vértices de control menos uno y la modificación de un vértice afecta a toda la curva, ya que las funciones $J_{n,i}(t)$ afectan globalmente a toda la curva. La base B-spline (de *Basis spline*) corrige estas limitaciones y es la que define las curvas B-spline.

Aunque el arquitecto o diseñador normalmente no necesita manejar la fórmula que define las curvas B-spline, tener una idea del papel que juegan los ingredientes necesarios para definirla puede ayudar a entender su funcionamiento. Más interesante que la fórmula, son las distintas formas de las curvas B-spline, que resultan de las funciones de la base B-spline, y que varían atendiendo a la elección de un vector de nudos y del grado k (en la literatura también se habla de orden igual a $k+1$), véase el ejemplo de la figura 2.1. Para definir una curva B-spline necesitamos:

1. Los vértices del polígono de control $B_0, \ldots, B_n$.
2. El grado k (orden $k+1$) menor o igual que n.
3. Un vector creciente de nudos $X = (a = x_1, \ldots, x_{n+k+2} = b)$, del que hablaremos en la Sección 2.1.

La parametrización de la curva B-spline está definida por [Rogers]

$$\mathcal{P}(t) = \sum_{i=1}^{n+1} B_{i-1} N_{i,k+1}(t), \quad a \leq t \leq b$$

con base de funciones B-spline $N_{1,l}, \ldots, N_{n+1,l}$, $l = 1, \ldots, k+1$ definidas de forma recursiva por las fórmulas de Cox-De Boor,

$$N_{i,1}(t) = \begin{cases} 1 & \text{si } x_i \leq t \leq x_{i+1}, \\ 0 & \text{en otro caso,} \end{cases}$$

$$N_{i,l+1}(t) = \frac{(t - x_i)N_{i,l}(t)}{x_{i+l} - x_i} + \frac{(x_{i+l+1} - t)N_{i+1,l}(t)}{x_{i+l+1} - x_{i+1}}, \quad l = 2, \ldots, k.$$

En esta definición tenemos que asumir que $0/0 = 0$. La curva B-spline está por tanto definida por una función spline polinomial que es de grado k y tiene derivadas continuas hasta orden $k-1$. Además $N_{i,l}(t) \geq 0$ y

$$\sum_{i=1}^{n+1} N_{i,l}(t) \equiv 1.$$

Observamos que la parametrización, al igual que en las curvas de Bézier, es polinomial y si $k < n$ está definida a trozos. Hay curvas, como las cónicas, que no admiten una parametrización polinomial y por tanto no se pueden parametrizar mediante curvas B-spline, hablaremos de ellas en la Sección 3. Las curvas de Bézier son un caso particular de curvas B-spline, con grado n, el máximo posible, y vector de nudos igual a

$$(0, \ldots^{(n+1)}, 0, 1, \ldots^{(n+1)}, 1).$$

La curva B-spline está formada por trozos de curvas de Bézier, todas ellas del mismo grado k, el grado de la curva B-spline. El vector de nudos contiene el valor del parámetro t en los puntos de unión de los trozos de curvas de Bézier.

Ejemplo 2.0.2. *Calculamos con Maple, la parametrización de la curva B-spline de grado* 2 *con puntos de control* $B_0 = (1,1)$, $B_1 = (3,5)$,$B_2 = (5,3)$,$B_3 = (7,4)$ *y vector de nudos* $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (0,0,0,1,2,2,2)$. *Es una parametrización polinomial a trozos*

$$\begin{aligned} \mathcal{P}(t) =& N_{1,3}(t)B_0 + N_{2,3}(t)B_1 + N_{3,3}(t)B_2 + N_{4,3}(t)B_3 \\ =& \begin{cases} (-t^2+4t+1, -5t^2+8t+1) & si\ 0 \leq t \leq 1, \\ (3+t^2, 8-6t+2t^2) & si\ 1 \leq t \leq 2, \end{cases} \end{aligned}$$

siendo $N_{i,3}(t)$ *las funciones de la gráfica de la figura 2.1. Observamos que la curva es la unión de dos curvas de Bézier, que se unen en el punto* $C = (4,4)$. *La curva de Bézier determinada por los puntos de control* B_0, B_1, C *tiene parametrización*

$$\mathcal{P}_1(t) = (1-t)^2B_0 + 2t(1-t)B_1 + t^2C = (-t^2+4t+1, -5t^2+8t+1), \ \ t \in [0,1].$$

La curva de Bézier determinada por los puntos de control C, B_2, B_3 *tiene parametrización*

$$\mathcal{P}_2(t) = (1-t)^2B_0 + 2t(1-t)B_1 + t^2C = (t^2+2t+4, 2t^2-2t+4), \ \ t \in [0,1].$$

Si reparametrizamos para que $t \in [1,2]$ *tenemos*

$$\mathcal{P}_2(t-1) = (3+t^2, 8-6t+2t^2), \ \ t \in [1,2].$$

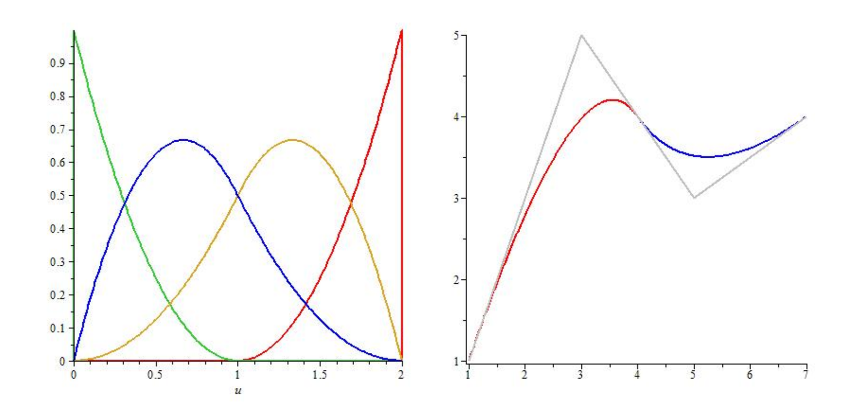


Figura 2.1: (Izqd) Funciones $N_{i,3}(t)$, $i = 1, \ldots, 4$ de grado 2 de la base B-spline y (dcha) curva B-spline de grado 2, para cuatro vértices

2.1. Vector de nudos

El vector de nudos X, es un lista creciente de números reales, con multiplicidad a lo sumo el grado $k+1$. La longitud de X depende del número de puntos de control $n+1$ y del orden $k+1$, siendo

$$X = [x_1, \ldots, x_{n+k+2}].$$

Hemos comentado que los valores que aparecen en el vector X son valores de los parámetros en los puntos de unión de los trozos de curvas de Bézier que forman la curva B-spline. Cuando los nudos están uniformemente espaciados se dice que la curva es B-spline uniforme y en otro caso no uniforme.

$$\begin{aligned}
&[1,2,3,4,5,6] \text{ uniforme para } n=2, k=2,\\
&[1,3,4,7,8,10] \text{ no uniforme},\\
&[1,2,2,3,4,4] \text{ periódico no uniforme}.
\end{aligned}$$

Los valores duplicados del nudo en la mitad de la lista hacen que una curva B-spline sea menos suave. Se dice que el vector de nudos es no uniforme periódico. En un caso extremo, un nudo de multiplicidad $k+1$ en la mitad de la lista de nudos significa que hay un lugar en la curva B-spline que se puede doblar en un punto de torsión. Por esta razón, a algunos diseñadores les gusta agregar y quitar nudos y luego ajustar los puntos de control para hacer curvas más suaves o figuras torsionadas.

Si una lista de nudos se inicia con un nudo de multiplicidad $k+1$, le siguen nudos simples y termina con un nudo de multiplicidad $k+1$ y los valores se espacian uniformemente, se dice que el vector de nudos es uniforme abierto (en inglés *open*

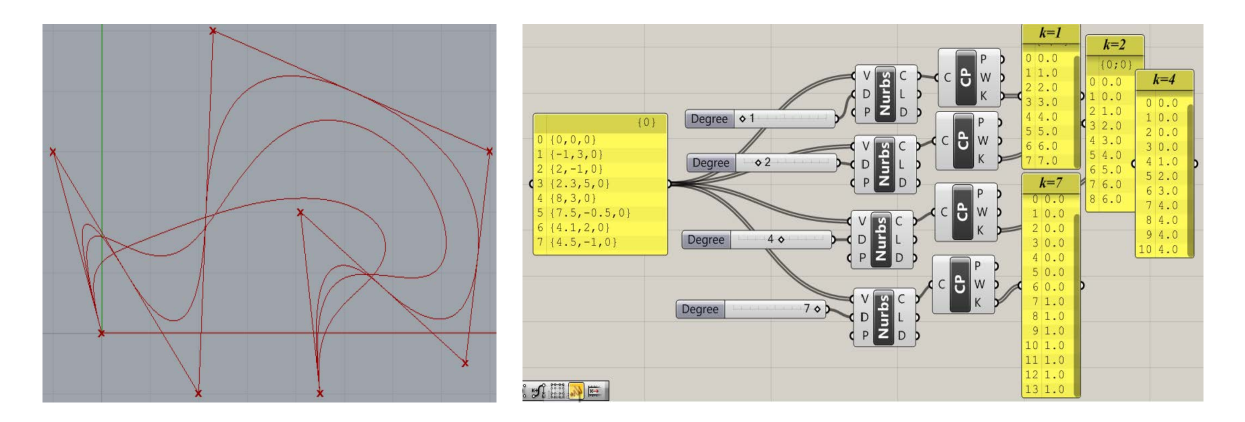


Figura 2.2: (Izq) Curvas B-spline de grados $k = 1, 2, 4, 7$. (Dcha) Código Grasshopper, haciendo visible el vector de nudos

uniform), pero en general no tiene por qué serlo.

$$
\begin{aligned}
&[0, 0, 1, 2, 3, 4, 4] \text{ uniforme abierto para } n = 4, k = 1,\\
&[0, 0, 0, 1, 2, 3, 3, 3] \text{ uniforme abierto para } n = 4, k = 2,\\
&[0, 0, 0, 1/3, 2/3, 1, 1, 1] \text{ normalizado al intervalo } [0, 1],\\
&[0, 0, 0, 0, 1, 2, 2, 2, 2] \text{ uniforme abierto para } n = 4, k = 3,\\
&[0, 0, 0, 1/2, 1, 1, 1, 1] \text{ normalizado al intervalo } [0, 1],\\
&[0, 0, 0, 0, 0, 1, 1, 1, 1, 1] \text{ uniforme abierto para } n = 4, k = 4.
\end{aligned}
$$

Se muestran en la figura 2.2 los vectores de nudos que toma Grasshopper para definir una curva B-spline de distintos grados. Observamos que en Grasshopper el vector de nudos tiene longitud $n + k$ y los nudos aparecen con multiplicidad a lo sumo k.

2.2. Influencia del grado

En general, una curva B-spline de grado k con $n+1$ puntos de control está formada por $n + 1 - k$ trozos de curva de Bézier de grado k. Fijado un polígono de control podemos obtener distintas curvas B-spline variando el grado. El grado mínimo es $k = 1$ y la curva B-spline es el polígono de control. Conforme aumenta el grado, la curva aproxima peor al polígono de control, (aunque aumenta la continuidad, véase la Sección 2.3). Observamos que modificar la longitud del vector de nudos es equivalente a cambiar el grado. Ilustramos estos fenómenos en la figura 2.2.

Las curvas B-spline nos permiten tener control local. Si el grado k es menor o igual que $n - 1$, al mover un punto de control, el cambio no afecta a toda la curva B-spline sino a un número de vértices adyacentes. En la figura 2.3 se muestran dos curvas que

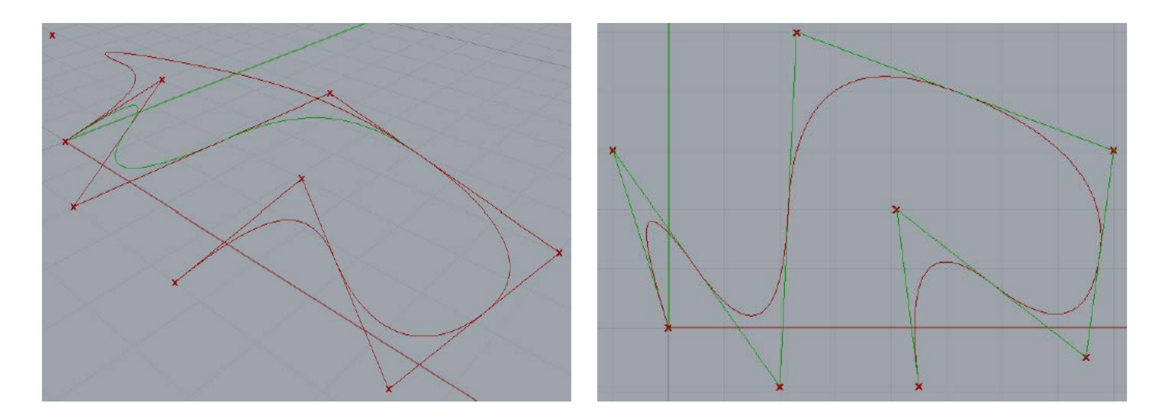


Figura 2.3: (Izq) Dos curvas B-spline de grado $k = 2$. (Dcha) Proyección al plano xy de ambas curvas

tiene todos sus puntos de control iguales excepto el B_2, que en la curva plana es igual a $(2, -1, 0)$ y en la alabeada es $(2, -1, 2)$. Las curvas son de grado $k = 2$ y tienen 8 puntos de control, por lo tanto están formadas por 6 curvas de Bézier $\mathcal{B}_1, \ldots, \mathcal{B}_6$ de grado 2. Los 5 puntos de intersección $C_1, \ldots, C_5$ de la curva plana con su polígono de control determinan los puntos de unión de las curvas de Bézier. La curva $\mathcal{B}_2$ tiene puntos de control C_1, B_2, C_2 y el desplazamiento del punto B_2 afecta también a las curvas adyacentes (afecta a $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$).

Las curvas B-spline admiten modo abierto y modo cerrado. Hasta ahora hemos utilizado el modo abierto, en el que la curva B-spline contiene los puntos inicial y final (B_0 y B_n) del polígono de control. En el modo cerrado, el polígono de control es cerrado, $B_0 = B_n$ y la curva B-spline es una curva lisa que (en general) no contiene a dichos puntos. En el ejemplo de la figura 2.4, ilustramos que el componente `Nurbs` de Grasshopper permite elegir modo cerrado, con entrada C igual a *true*. En modo cerrado tendríamos 9 puntos de control, ya que el primero se repite al final. En este caso hemos tomado grado 6. Observamos como la curva abierta tiene un pico en el punto $B_0 = B_8$ y sin embargo la curva cerrada es suave.

Por estar formadas por una unión de curvas de Bézier, las curvas B-spline heredan propiedades de estas:

1. Al igual que las curvas de Bézier, las curvas B-spline están contenidas en la envolvente convexa de los puntos de control.

2. El ejemplo de la figura 2.3 ilustra que las curvas B-spline también son invariantes mediante transformaciones afines, en este caso observamos como la proyección de la curva de Bézier espacial sobre el plano xy, coincide con la curva B-spline de los puntos de control proyectados.

3. La parametrización es de clase C^{k-1} (en los nudos).

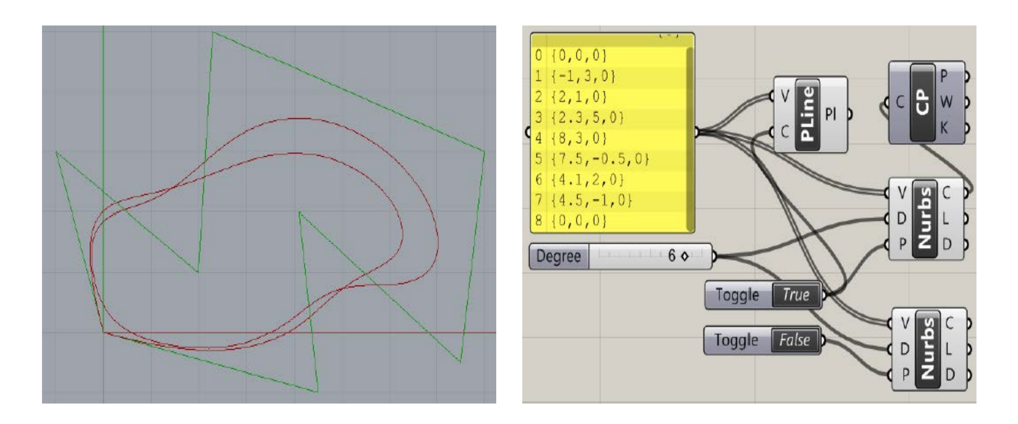


Figura 2.4: (Izq) Dos curvas B-spline de grado $k = 2$, una abierta (en verde) y otra cerrada. (Dcha) Código Grasshopper

2.3. Continuidad geométrica

Sea Γ la unión de dos (trozos de) curvas Γ_1 y Γ_2 en un punto extremo P común a ambas. Analicemos la continuidad de Γ en el punto P.

1. Por unirse en un punto extremo común P, el segmento Γ es continuo en P, diremos que es $\mathcal{C}^0$ (la continuidad clásica). También se dice que Γ tiene continuidad geométrica G^0 en P. Las nociones de continuidad $\mathcal{C}^0$ y G^0 coinciden.

2. Si los vectores tangentes v_1 a Γ_1 y v_2 a Γ_2 en P tienen la misma dirección $v_1 = \alpha v_2$, $\alpha > 0$, diremos que Γ tiene continuidad geométrica G^1 en P. Observamos que si la parametrización $\mathcal{P}(t)$ de Γ es de clase $\mathcal{C}^1$ en P entonces los vectores tangentes tienen la misma dirección y la misma magnitud, por lo que $\mathcal{C}^1$ es más exigente que G^1. Así, $\mathcal{C}^1$ implica G^1 pero no al revés.

3. Teniendo continuidad geométrica G^1 en P, diremos que Γ tiene continuidad geométrica G^2 si los vectores curvatura κ_1 y κ_2 en P de Γ_1 y Γ_2 tienen la misma dirección $\kappa_1 = \alpha\kappa_2$, $\alpha > 0$.

Aunque en muchas aplicaciones es suficiente exigir continuidad geométrica G^1, existen otras (aquellas que dependen de la suavidad de las curvas o superficies) en las que es necesaria una continuidad G^2 o incluso C^2, especialmente las que dependen de una transición suave de la luz reflejada, como por son por ejemplo los automóviles [Rogers].

Ejemplo 2.3.1. *Conectamos dos curvas Γ_1 y Γ_2 de Bézier mediante el componente* `Connect` *de Grasshopper, que toma como input una lista de curvas $C = \{\Gamma_1, \Gamma_2\}$ y un número entero G, la continuidad geométrica $0, 1$ ó 2. Las curvas has sido creadas con*

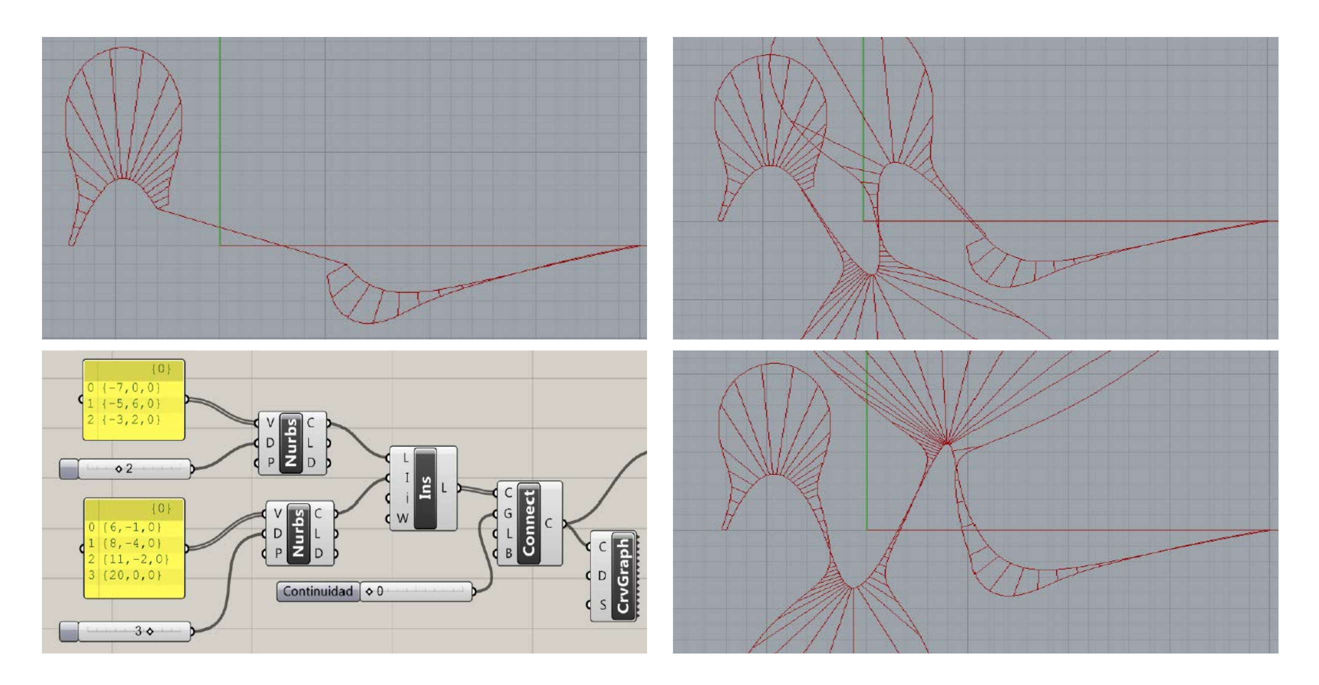


Figura 2.5: Curvas conectadas con continuidad (arriba izq) G^0, (arriba dcha) G^1, (abajo dcha) G^2. (Abajo izq) código Grasshopper

el componente **`Nurbs`**, *la primera de grado* 2 *(con* 3 *puntos de control) y la segunda de grado* 3 *(con* 4 *puntos de control). Utilizamos el componente* **`CrvGraph`** *para visualizar el gráfico de curvatura de la curva resultado* Γ*. El componente* **`Ins`** *nos permite insertar las curvas* Γ_1 *y* Γ_2 *en una misma lista. En la figura 2.5 mostramos el resultado de conectar las curvas utilizando los distintos tipos de continuidad geométrica* G^0, G^1 *y* G^2.

Es interesante observar que, para conectar Γ_1 *y* Γ_2 *con continuidad* G^1, *Grasshopper genera una curva B-spline de grado* 3 *con vector de nudos periódico*

$$X = [0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3].$$

Si le pedimos continuidad G^2, *genera una curva B-spline de grado* 5 *con vector de nudos*

$$X = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3].$$

3 Curvas B-spline racionales (NURBS)

Ya hemos hablado de casi todos los ingredientes de las curvas NURBS (*Non uniform rational B-spline*), nos falta explicar qué significa ser racional. Es preciso matizar que el nombre NURBS no es excluyente, incluye a todos los casos anteriormente estudiados. Son curvas NURBS las curvas de Bézier y las curvas B-spline, tanto uniformes como no uniformes. El aspecto racional es el que nos permitirá dibujar todas las cónicas mediante este tipo de curvas, véanse los detalles de la Sección 3.1.

Una curva NURBS plana es la proyección central de una curva B-spline espacial sobre un plano. Análogamente una curva NURBS espacial es la proyección central, sobre un hiperplano, de una curva B-spline que vive en un espacio de dimensión 4. De forma más precisa, sean $B_0, \ldots, B_n$, puntos de control en $\mathbb{R}^3$, supongamos que $B_i = (a_i, b_i, c_i)$. Tomemos coordenadas homogéneas $B_i^h = w_i(1, a_i, b_i, c_i) \in \mathbb{R}^4$, $w_i > 0$ de B_i. Una curva B-spline racional espacial es la curva afín de una curva B-spline proyectiva

$$\widetilde{\mathcal{P}}(t) = \sum_{i=1}^{n+1} B_{i-1}^h N_{i,k+1}(t), \quad a \le t \le b.$$

Pasando a coordenadas afines obtenemos la parametrización de la curva afín, la curva B-spline racional

$$\mathcal{P}(t) = \frac{\sum_{i=1}^{n+1} w_{i-1} B_{i-1} N_{i,k+1}(t)}{\sum_{i=1}^{n+1} w_{i-1} N_{i,k+1}(t)}, \quad a \le t \le b. \tag{3.1}$$

Llamamos peso del punto de control B_i a la primera coordenada w_i de B_i^h. Nos restringimos a pesos positivos $w_i > 0$ para evitar (en este manuscrito) utilizar puntos en el infinito. El incremento del valor del peso w_i acerca la curva NURBS al punto de control B_i, si el valor del peso decrece se produce el efecto contrario y la curva NURBS se aleja del punto de control. Este desplazamiento tiene un efecto local en la curva NURBS.

Las curvas NURBS que tienen todos sus pesos iguales, $w_i = \omega$, $i = 0, \ldots, n$ son curvas B-spline, la parametrización (3.1) es polinomial ya que el denominador es

constante

$$\sum_{i=1}^{n+1} w_{i-1} N_{i,k}(t) = \omega \sum_{i=1}^{n+1} N_{i,k}(t) = \omega,$$

debido a que la suma de las funciones B-spline $N_{i,k}(t)$, $i = 1, \ldots, n+1$ es igual a 1. En este caso, la curva parametrizada por $\widetilde{\mathcal{P}}(t)$ sería una curva B-spline en el hiperplano $w = \omega$, semejante a su proyección en el hiperplano $w = 1$, parametrizada por $\mathcal{P}(t)$. En otro caso, si existe algún peso distinto de los demás, la parametrización $\mathcal{P}(t)$ es racional y tenemos una NURBS genuina, que no es B-spline.

El diseñador tiene que tener en cuenta que para modelar utilizando curvas NURBS, necesita establecer el valor de:

1. Los puntos de control $B_0, \ldots, B_n$.

2. El grado k menor o igual que n y el vector de nudos X.

3. Los pesos $w_i > 0$, $i = 0, \ldots, n$.

Podría jugar con todos estos datos hasta obtener la curva deseada. Si utiliza curvas B-spline, sólo podría modificar puntos de control, grado y vector de nudos, los pesos son todos iguales a 1. El caso menos versátil es el de las curvas de Bézier, para las que sólo podría modificar los puntos de control, el grado es fijo (número de puntos de control menos uno, y el vector de nudos también) y los pesos son todos iguales a 1.

El componente de Grasshopper que nos permite controlar todos los ingredientes de una curva NURBS es `NurbCrv`, que toma como entradas:

1. P la lista de puntos de control $B_0, \ldots, B_n$.

2. W la lista de pesos $w_0, \ldots, w_n$.

3. K el vector de nudos X.

El grado de la curva NURBS queda determinado por la longitud l del vector de nudos, que en Grasshopper es $l = n + k$ (aunque en la definición de la base B-spline es $n + k + 2$). Como ya comentamos en la Sección 2.1, modificar el vector de nudos equivale a elegir el grado.

En el ejemplo de la figura 3.1 ilustramos el efecto de cambiar el valor del peso w_1. La curva de partida es una circunferencia del plano xy, de radio 20 y centrada en el origen. Observamos que es una curva B-spline racional, formada por 4 arcos de curva de grado 2 (cada arco sería una curva de Bézier racional). El arco del primer cuadrante está determinado por los puntos B_0, B_1 y B_2.

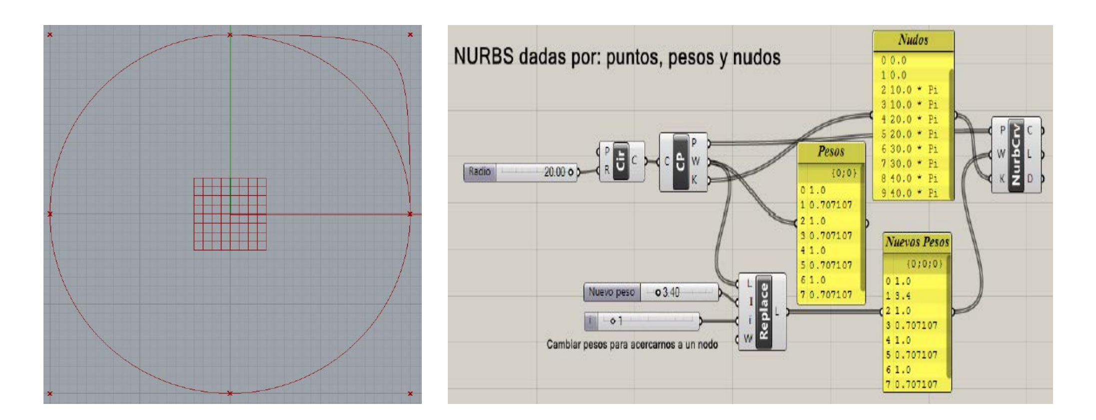


Figura 3.1: (Izq) Una circunferencia y la curva NURBS que resulta al cambiar el peso de un punto de control. (Dcha) Código Grasshopper para modificar puntos, pesos y nudos en una curva NURBS, en el componente `NurbCrv`

3.1. Cónicas

Las secciones cónicas son curvas descritas por ecuaciones de segundo grado, que además tienen parametrizaciones racionales. Por este motivo es interesante analizar las curvas NURBS dadas por 3 puntos de control B_0, B_1, B_2 y con grado máximo $k = 2$. Veremos que son arcos de cónica, y que sólo las parábolas tienen parametrizaciones polinomiales.

Supongamos que trabajamos con puntos en el plano $B_i = (a_i, b_i)$, $i = 0, 1, 2$. Elegimos coordenadas homogéneas de dichos puntos

$$B_i^h = (w_i, w_i a_i, w_i b_i), i = 0, 1, 2,$$

con pesos $w_i > 0$. Los puntos B_0^h, B_1^h y B_2^h determinan una curva de Bézier $\tilde{\mathcal{C}}$ de grado 2 espacial, parametrizada por

$$\begin{aligned}\tilde{\mathcal{P}}(t) =& \sum_{i=0}^{2} J_{2,i}(t) B_i^h = \\ =& (1-t)^2 B_0^h + 2(1-t)t B_1^h + t^2 B_2^h = \\ =& (\mathcal{P}_0(t), \mathcal{P}_1(t), \mathcal{P}_2(t)).\end{aligned}$$

La curva $\tilde{\mathcal{C}}$ es una cónica proyectiva, cuya cónica afín $\mathcal{C}$ está parametrizada por

$$\mathcal{P}(t) = \left(\frac{\mathcal{P}_1(t)}{\mathcal{P}_0(t)}, \frac{\mathcal{P}_2(t)}{\mathcal{P}_0(t)} \right) = \frac{\sum_{i=0}^{2} w_i J_{2,i}(t) B_i}{\sum_{i=0}^{2} w_i J_{2,i}(t)}.$$

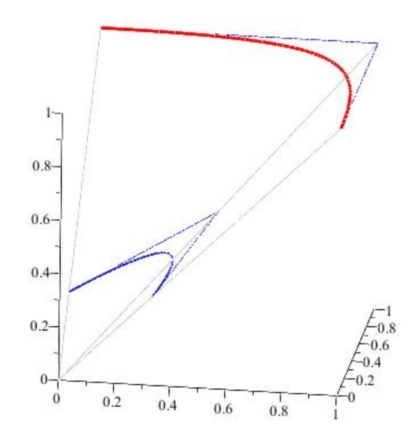


Figura 3.2: Curva proyectiva y su proyección sobre el plano afín ($x_0 = 1$)

Ejemplo 3.1.1. *Tomemos puntos de control $B_0 = (0,1)$, $B_1 = (1,1)$, $B_2 = (1,0)$ y pesos $w_0 = \frac{1}{3}$, $w_1 = \frac{1}{2}$, $w_2 = \frac{1}{4}$. La parametrización de la cónica proyectiva (curva de Bézier de grado 2) es*

$$\tilde{\mathcal{P}}(t) = \left(-\frac{5}{12}t^2 + \frac{1}{3}t + \frac{1}{3}, \frac{1}{3}t - \frac{2}{3}t^2 + \frac{1}{3}, -\frac{3}{4}t^2 + t\right).$$

Representamos en la figura 3.2 la cónica proyectiva $\tilde{\mathcal{C}}$ y su proyección central $\mathcal{C}$ sobre el plano afín (si denotamos por (x_0, x_1, x_2) las coordenadas de un punto del plano proyectivo, estamos proyectando sobre el plano $x_0 = 1$).

Observamos que la primera coordenada de $\tilde{\mathcal{P}}(t)$ es $\mathcal{P}_0(t) = \sum_{i=0}^{2} J_{2,i}(t)w_i$ y los puntos del infinito de $\tilde{\mathcal{C}}$ son aquellos para los que $\mathcal{P}_0(t) = 0$. Es conveniente tomar

$$w_0 = w_2 = 1 \text{ y } w_1 = w.$$

Estudiemos cuántos puntos del infinito tiene $\tilde{\mathcal{C}}$ en función del valor de w, lo que equivale a determinar el tipo de cónica. Las soluciones de

$$\mathcal{P}_0(t) = (1-t)^2 + 2(1-t)tw + t^2 = 0.$$

son

$$t_1 = \frac{-1 + w + \sqrt{-1 + w^2}}{2(w-1)} \text{ y } t_2 = -\frac{1 - w + \sqrt{-1 + w^2}}{2(w-1)}.$$

Fácilmente observamos que la cónica proyectiva $\tilde{\mathcal{C}}$ tiene los siguientes puntos en el infinito:

1. Si $w = 1$ entonces $t_1 = t_2$, un punto doble en el infinito. La cónica afín $\mathcal{C}$ es una parábola.

2. Si $0 < w < 1$ entonces t_1 y t_2 son valores complejos, dos puntos del infinito complejos. La cónica afín $\mathcal{C}$ es una elipse.

3. Si $w > 1$ entonces t_1 y t_2 son valores reales, dos puntos del infinito reales. La cónica afín $\mathcal{C}$ es una hipérbola.

Podemos calcular la ecuación implícita $f(x, y) = 0$ de la cónica afín eliminando la variable t de los polinomios

$$\mathcal{P}_0(t)x - \mathcal{P}_1(t) = 0,$$
$$\mathcal{P}_0(t)y - \mathcal{P}_2(t) = 0.$$

Para ello podemos utilizar el comando Maple, `resultant`. Por ejemplo, si tomamos $B_0 = (0, 1)$, $B_1 = (1, 1)$ y $B_2 = (1, 0)$

$$\mathcal{P}_1(t) = 2wt - 2wt^2 + 1 - 2t + t^2,$$
$$\mathcal{P}_2(t) = t^2 + 2wt - 2wt^2.$$

y así

$$\begin{aligned} f(x, y) =& \texttt{resultant}(\mathcal{P}_0(t)x - \mathcal{P}_1(t), \mathcal{P}_0(t)y - \mathcal{P}_2(t), t) \\ =& 1 - 2y - 2x - 4w^2 + y^2 + x^2 + 2yx + 4xw^2 - 4yw^2x + 4yw^2 = \\ =& \begin{pmatrix} 1 & x & y \end{pmatrix} \begin{pmatrix} 1 - 4w^2 & -1 + 2w^2 & -1 + 2w^2 \\ -1 + 2w^2 & 1 & 1 - 2w^2 \\ -1 + 2w^2 & 1 - 2w^2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}. \end{aligned}$$

Observamos que la matriz de la cónica es degenerada sólo cuando $w = 0$. Además si $w = \frac{\sqrt{2}}{2}$ entonces $\mathcal{C}$ es el arco de la circunferencia $x^2 + y^2 = 1$ con parametrización

$$\mathcal{P}(t) = \left(\frac{t^2 + t\sqrt{2} - t^2\sqrt{2}}{2t^2 + t\sqrt{2} - t^2\sqrt{2} + 1 - 2t}, \frac{t\sqrt{2} - t^2\sqrt{2} + 1 - 2t + t^2}{2t^2 + t\sqrt{2} - t^2\sqrt{2} + 1 - 2t} \right)$$

Ejemplo 3.1.2. *Tomemos puntos de control $B_0 = (0, 0)$, $B_1 = (3, 4)$, $B_2 = (4, 0)$ y pesos $w_0 = 1$, $w_1 = w$, $w_2 = 1$. Representamos en la figura 3.3 tres arcos de cónica para distintos valores de w. Si $w = \frac{1}{4}$ es un arco de elipse, $w = 1$ un arco de parábola y $w = 3$ un arco de hipérbola. Las parametrizaciones son:*

$$\left(\frac{t^2 + 3t}{t^2 - t + 1}, \frac{4t - 4t^2}{t^2 - t + 1} \right) \text{ para } w = \frac{1}{4},$$
$$\left(-2t^2 + 6t, 8t - 8t^2 \right) \text{ para } w = 1,$$
$$\left(\frac{-14t^2 + 18t}{-4t^2 + 4t + 1}, \frac{24t - 24t^2}{-4t^2 + 4t + 1} \right) \text{ para } w = 3.$$

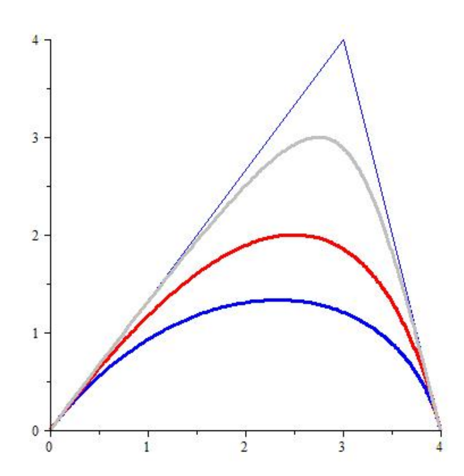


Figura 3.3: Cónicas para (de abajo a arriba) $w = \frac{1}{4}$, $w = 1$ y $w = 3$

Nótese que sólo la parábola tiene parametrización polinomial.

Bibliografía

[CLl] M. Castellet, I. Llerena. Álgebra Lineal y Geometría. Ed. Reverté, 2009.

[GAS] A. Gray, E. Abbena, S. Salamon. Modern Differential Geometry of Curves and Surfaces with Mathematica. Studies in Advanced Mathematics. Chapman & Hall, 3rd Ed, 2006.

[PAHK] H. Pottmann, A. Asperl, M. Hofer, A. Kilian. Architectural Geometry. Bentley Institute Press, 2007.

[Rogers] D.F. Rogers. An introduction to NURBS. With historical perspective. Academic Press, 2001.

NOTAS

NOTAS

CUADERNO



Cuadernos.ijh@gmail.com
info@mairea-libros.com

